UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/637,015	08/14/2000	Gopal S. Krishna	95-319	8307

20736 7590 12/20/2004

MANELLI DENISON & SELTER
2000 M STREET NW SUITE 700
WASHINGTON, DC 20036-3307

EXAMINER	
MOORE, IAN N	
ART UNIT	PAPER NUMBER
2661	

DATE MAILED: 12/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

12

Office Action Summary	Application No.	Applicant(s)
	09/637,015	KRISHNA ET AL.
	Examiner	**Art Unit**
	Ian N Moore	2661

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 July 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's request for reconsideration of the finality of the rejection of the last Office action and arguments appeal brief is persuasive and, therefore, the finality of that action is withdrawn.

2. Claims 1-16 are rejected with new grounds of rejection.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1 and 11 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 8, 14, 1 and 7 of U.S. Patent No. 6,693,906. Although the conflicting claims are not identical, they are not patentably distinct from each other because claims 1 and 11 of the instant application merely broadens the scope of the claims 8 and 11 of the Patent by eliminating the elements and their functions (i.e. by changing plurality of equation cores to one equation core) of the claims. It has been held that the omission an element and its function is an obvious expedient if the remaining elements perform the same function as

before. *In re Karlson*, 136 USPQ 184 (CCPA). Also note *Ex parte Rainu*, 168 USPQ 375 (Bd.App.1969); omission of a reference element whose function is not needed would be obvious to one skilled in the art.

5. Claims 1 and 11 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1,4, and 14 of U.S. Patent No. 6,700,897. Although the conflicting claims are not identical, they are not patentably distinct from each other because claims 1 and 11 of the instant application merely broadens the scope of the claims 1,4, and 14 of the Patent by adding the <u>inherent</u> functions (i.e. comparing upon receipt) of the claims.

6. Claims 1 and 11 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1,4, 13 of U.S. Patent No. 6,741,594. Although the conflicting claims are not identical, they are not patentably distinct from each other because claims 1 and 11 of the instant application merely broadens the scope of the claims 8 and 11 of the Patent by eliminating the elements and their functions (i.e. first and second branches) and adding inherent functions (i.e. comparing upon receipt) of the claims. It has been held that the omission an element and its function is an obvious expedient if the remaining elements perform the same function as before. *In re Karlson*, 136 USPQ 184 (CCPA). Also note *Ex parte Rainu*, 168 USPQ 375 (Bd.App.1969); omission of a reference element whose function is not needed would be obvious to one skilled in the art.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 1-10 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites, "simultaneously comparing" in line 6, "generating a comparison result" in line 9, and "generation a frame tag **based** on the comparison result **as soon as a last bit** of the data packet is received" in line 12-13. It is unclear when a frame tag is generated. Per claimed invention, the generation a frame tag depends on two preconditions **1)** based on the comparison result **2)** as soon as a last bit is received. However, the simultaneous or parallel comparing process occurs until the last bit of the packet is received. Consequently, the comparison result can only be generated only **after** simultaneous or parallel comparing process of entire packet. Thus, it is unclear when a frame tag is generated since precondition 2 cannot be met.

Claims 2-10 are also rejected since they depended on rejected claim 1.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1-3,5,6,8,9, 10 and 15 are rejected under 35 U.S.C. 102(e) as being anticipated by Bellenger (U.S. 5,949,786)

Regarding Claim 1, Bellenger discloses a method of evaluating an incoming data packet at a network switch port (see FIG. 2, Network Router port 101; see col. 4, lines 50-55), the method comprising:

storing a plurality of templates (see FIG. 2, the flow detect logic 105 loaded-with/includes a plurality of filters (i.e. template registers 600, protocol filter 601, hierarchy filter602 of FIG. 6); see col. 7, lines 15-35) configured for identifying respective data formats (see col. 7, lines 15-19, 24-30), each template having at least one min term (see col. 9, lines 50 col. 10, lines 40; one byte min term (i.e. FF, F0, or 00) in template register) configured for comparing (see FIG. 5, parallel simultaneous comparison/matching analysis process for each hash flow byte of the frame by the filters; see col. 6, lines 22-30; see col. 6, lines 60-61) a corresponding prescribed value (see col. 9, lines 50 to col. 10, lines 40; selected/used byte from a template) to a corresponding selected byte of the incoming data packet (see FIG. 4, selected byte EST 410, Source 411, IP header of received frame; see col. 9, lines 50 to col. 10, lines 40;

simultaneously comparing to the selected byte (see col. 6, lines 22-30; see col. 6, lines 60-61), the min terms (see col. 9, lines 50 col. 10, lines 40; plurality of min terms in the template register (i.e. FF, F0, 00, F2, 03, FC)) that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port (see FIG. 6, start address 603; see col. 6, lines 22-30; see col. 6, lines 60-61; see col. 10, lines 45-50; note that matching/hashing process is performed on-the-fly upon received of a frame);

generating a comparison result (see FIG. 5, Hash value 503) that identifies the incoming data packet (see col. 7, lines 36-38, 55-56; hash value is generated for selected flow

of the incoming frame), based on the comparisons of the min terms to the data bytes of the entire packet received by the network switch port (see col. 6, lines 30-34, see col. 7, lines 16-65); and

generating a frame tag (see FIG. 6, Tag) based on the comparison result (see FIG. 6, hash value 503) as soon as a last bit of the data packet is received (see FIG. 6, received data frame 500) at the network switch port (see col. 7, lines 60 to col. 8, lines 6).

Regarding Claim 11, Bellenger discloses a network switch port filter configured for evaluating an incoming data packet (see FIG. 2, Network Router port 101; see col. 4, lines 50-55), comprising:

a min term memory (see FIG. 2, the flow detect logic 105 loaded-with/includes/stores a plurality of filters (i.e. template registers 600, protocol filter 601, hierarchy filter602 of FIG. 6); see col. 7, lines 15-35) configured for storing (see col. 9, lines 50 col. 10, lines 40; min term values (i.e. FF, F0, 00) in template register), each min term value stored based on a location of a corresponding selected byte of the incoming data packet (see FIG. 4, selected byte EST 410, Source 411, IP header of received frame; see col. 9, lines 50 to col. 10, lines 40) for comparison (see FIG. 5, parallel simultaneous comparison/matching analysis process for each hash flow byte of the frame by the filters; see col. 6, lines 22-30; see col. 6, lines 60-61), a min term portion specifying a corresponding comparison operation (see col. 9, lines 50 to col. 10, lines 40; used fields in the templates for IP or IPX), and an equation identifier field that specifies templates that use the corresponding min term (see col. 9, lines 50 to col. 10, lines 40; IP or IPX filter/template that specifies different used byte fields);

a min term generator (see FIG. 6, a combined system of Mask 617, Match 614, Psuedo-random number gen 620, and has result 604, or see FIG. 5, a combined system of hash flows 1-N and multiplexed 503) configured for simultaneously comparing a byte of the incoming data packet (see col. 6, lines 22-30; see col. 6, lines 60-61) immediately upon receipt of the incoming data byte (see FIG. 6, start address 603; see col. 6, lines 22-30; see col. 6, lines 60-61; see col. 10, lines 45-50; note that matching/hashing process is performed on-the-fly upon received of a frame), with the min terms (see col. 9, lines 50 col. 10, lines 40; plurality of min terms in the template register (i.e. FF, F0, 00, F2, 03, FC)) that corresponds to the received byte and generating respective min term comparison results (see FIG. 5, Hash value 503; see col. 7, lines 36-38, 55-56; hash value is generated for selected flow of the incoming frame).

an equation core (see FIG. 6, Routing Table 632) configured for generating a frame tag (see FIG. 6, Tag) identifying the incoming data packet based on the min term comparison results (see FIG. 6, hash value 503) relative to the templates (see col. 7, lines 60 to col. 8, lines 6).

Regarding Claim 2, Bellenger discloses loading the min terms corresponding to a first of the data bytes (see col. 7, lines 15-35; see col. 9, lines 30 to col. 10, lines 40; see FIG. 4, Destination address 410) into a min term generator (see FIG. 6, a combined system of Mask 617, Match 614, Psuedo-random number gen 620, and has result 604, or see FIG. 5, a combined system of hash flows 1-N and multiplexed 503), comparing in the min terms loaded in the min term generator with the first of the data bytes (see FIG. 5, parallel simultaneous comparison/matching analysis process for each hash flow byte of the frame by

the filters; see col. 6, lines 22-30; see col. 6, lines 60-61); and outputting comparison results (see FIG. 5, Hash value 503; see col. 7, lines 36-38, 55-56) for the min terms loaded in the min term generator (see col. 9, lines 50 col. 10, lines 40; plurality of min terms in the template register (i.e. FF, F0, 00, F2, 03, FC).

Regarding Claim 3, Bellenger discloses loading the min terms corresponding to a second of the data bytes (see FIG. 4, Source address 411; see col. 9, lines 54-57), contiguously following the first of the data bytes (see FIG. 4, Destination address 410; see col. 9, lines 54-57; see col. 5, lines 40-65), into the min term generator (see col. 9, lines 30-50; see col. 7, lines 10-25).

Regarding Claim 5, Bellenger discloses the storing step includes storing each term in a memory as a table entry (see col. 9, lines 50 to col. 10, lines 40; template registers table/entries), each table entry having a location in the memory (see col. 9, lines 50 to col. 10, lines 40; destination address, frame address, etc. location in the template) based on a location of the corresponding selected byte in the incoming data packet (see FIG. 4, destination address 410, frame address 411, etc. of received frame), the table entry including a term expression portion specifying the corresponding prescribed value (see FIG. 4, the values destination address 410, frame address 411, etc. of received frame) and a comparison operator field (see col. 9, lines 50 to col. 10, lines 40; the destination address, frame address, in the template), and an evaluation portion having an equation identifier field that specifies the templates that use the corresponding min term (see col. 9, lines 50 to col. 10, lines 40; IP or IPX filter/template that specifies different used byte fields).

Regarding Claim 6, Bellenger discloses temporarily storing results of the comparisons of the terms to the selected bytes of the incoming data packet (see FIG. 4, Heash generator 414, or see FIG. 6, Hash result 604; see col. 5, lines 55-60; see col. 7, lines 35-40) detecting at least one matched template from the plurality of templates based on the results of the comparisons of the terms (see col. 5, lines 55-60; see col. 7, lines 35-40) and generating the comparison result based on the detected at least one matched template (see FIG. 4, hash value 415, see FIG. 6, hash value; see col.7, lines 25-67.)

Regarding Claim 8, Bellenger discloses the first of the data bytes (see col. 9, lines 50-60; a destination address-used) corresponds to a first of the data bytes of a packet having a prescribed format (see FIG. 4, destination address 410 of the frame), the simultaneously comparing step including evaluating (see FIG. 5, comparing/matching/hashing hash flow 1-N; see FIG. 6, match 614, mask 617, psuedo-random number gen 620, and hierarchy 602) the selected data byte (see col. 9, lines 50-60; a destination address-used)) relative to a beginning of the packet having the prescribed format (see FIG. 4, destination address 410 of the frame).

Regarding Claim 9, Bellenger discloses the prescribed format is Internet protocol (IP) format (see col. 10, lines 1-20; IP format).

Regarding Claim 10, Bellenger discloses identifying for each of the min terms compared with the incoming data packet a corresponding equation (see col. 9, lines 64-65; see col. 10, lines 16-20; a template register equation (i.e. FF-F0-00-00-00-00-00-00 for bridging)) and each equation specifying a unique result for a selected group of the templates; and generating the comparison result by having the detected at least one matched template (see col. 7, lines 10 to col. 8, lines 20).

Regarding Claim 15, Bellenger discloses the equation core generates the frame tag at a wire rate of the incoming data packet and prior to an end of the incoming data packet (see col. 10, lines 45-55; processing on-the-fly).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 4 and 12-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bellenger in view of Deb (U.S. 6,172,990).

Regarding Claim 4, Bellenger discloses outputting the frame tag as described above in claim 1. Bellenger does not explicitly disclose a switch fabric configured for selectively switching the incoming data packet based on the corresponding frame tag. However, Deb teaches outputting the frame tag to a switch fabric (see Fig. 2A, a combined system of Tx micro-RISC Stream Processor 114a and Switch Table Lookup 806) configured for selectively switching the incoming data packet based on the corresponding frame tag (see col. 21, line 14-61; and col. 22, line 46-57; noted that the user defined appendix is append to an incoming packet, transferred to a lookup switch table). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a switch fabric to switch the frame tags, as taught by Deb in the system of Bellenger, so that it would provide programmable processor for processing receive and transmit data over a high

speed network, and reduce transmission delay, see Deb col. 3, line 60-65, see col. 4, lines 55-65.

Regarding Claim 12, Bellenger discloses outputting the frame identifier, the selected byte of the incoming packet determined based on the identified packet (see col. 9, lines 50 to col. 10, lines 40) and as described above in claim 11. Bellenger does not explicitly disclose identifying a type of layer 2 packet. However, Deb discloses a frame identifier (see Fig. 3B, CAM 334) configured for identifying a type of layer 2 packet (see col. 9, line 18-23; and col. 13, line 16-23; noted that MAC layer is a Layer 2, and the CAM identifies by utilizing a look up table to process for each word type before passing over to the higher layers), the selected byte of the incoming data packet determined based on the identified type of layer 2 packet (see col. 13, line 59-63; col. 19, line 59 to col. 20, line 3; and Fig. 5A and 5B; noted that selected word of the incoming packet is determined according to the user defined instructions for different type of layer 2.) Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to identify type of layer 2 packet, as taught by Deb in the system of Bellenger, so that it would provide programmable processor for processing receive and transmit data over a high speed network, and reduce transmission delay, see Deb col. 3, line 60-65, see col. 4, lines 55-65.

Regarding Claim 13, Bellenger discloses location of each stored term value is relative to a beginning of an IP frame (see col. 10, lines 1-40), and as recited in claim 11 above. Deb discloses the location of each stored term value is relative to a beginning of an IP frame (see Fig.9, IP header 910; and col. 20, line 54-60; noted that received packet header is an IP header, and therefore, it is an IP frame.) within the layer 2 packet (see col. 13, line 33-

50; the user defined instructions are resident in word count 308, and it is configured to identify a desired word count in an in-coming packet. Each time a new packet is received by micro-RISC stream processor 114a, a word counter 307 will reset to "0", and then word counter 307 begins sequentially counting each word that is received into pipeline register stages 323 from data path 115a. Therefore, each stored the user defined instruction is relative to a beginning of an IP frame). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to identify type of layer 2 packet, as taught by Deb in the system of Bellenger, so that it would provide programmable processor for processing receive and transmit data over a high speed network, and reduce transmission delay, see Deb col. 3, line 60-65, see col. 4, lines 55-65.

Regarding Claim 14, Bellenger discloses a min term controller (see FIG. 2, Hash generators and protocol detect logic 105) configured for fetching the min terms from the min term memory corresponding to a selected byte of the IP frame within the incoming data packet (see col. 4, lines 55-65; see col. 5, lines 40-60; see col. 6, lines 20-35; see col. 7, lines 11-40). Deb discloses a min term controller (see Fig. 3B, Execution Logic 312) configured for fetching the min terms from the min term memory corresponding to a selected byte of the IP frame within the incoming data packet (see col. 15, line 45-49 and col.14, line 12-19; noted that the execution logic unit is preferably designed to control the examination of the received packet by the analyzing computer.)

13. Claim 7 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bellenger in view of Deb, as described above in claims 1 and 11, and further in view of Connery (U.S. Patent 6,570,884).

Regarding claim 7, the combined system of Bellenger and Deb discloses all of the limitations as recited in claim 1 above. Neither Deb nor Sarkissian explicitly disclose resolving a priority of templates to one final template when more than one template matches the incoming data packet. However, Connery discloses resolving a priority of templates (see Fig. 3, Pattern Match units 1-4) to one final template when more than one template matches the incoming data packet (see col. 7, line 52-62; noted that when there are multiple matching of patterns (i.e. more than one matching to the defined pattern), the processor determines the final matching format). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide hardware pattern matching logic that resolve a priority of templates to one final template, as taught by Connery, in the combined system of Bellenger and Deb, so that it would provide hardware pattern matching logic which supports pattern matching at the speed of the incoming packet stream, and signals the embedded processor when a packet having one of the plurality of variant formats is detected, thereby, minimizes the probability of faulty matches; see Connery col. 3, line 46-53.

Regarding claim 16, the combined system of Bellenger and Deb discloses all of the limitations as recited in claim 11 above. Neither Bellenger nor Deb explicitly disclose a priority device configured for resolving a priority of templates to one final frame template when more than one template matches the incoming data packet. However, Deb teaches a priority device (see Fig. 3, Processor 220) configured for resolving a priority of templates

(see Connery '884 Fig. 3, Pattern Match units 1-4) to one final frame template when more than one template matches the incoming data packet (see col. 7, line 52-62; noted that when there are multiple matching of patterns (i.e. more than one matching to the defined pattern), the processor determines the final matching format). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide hardware pattern matching logic that resolve a priority of templates to one final template, as taught by Connery, in the combined system of Bellenger and Deb, so that it would provide hardware pattern matching logic which supports pattern matching at the speed of the incoming packet stream, and signals the embedded processor when a packet having one of the plurality of variant formats is detected, thereby, minimizes the probability of faulty matches; see Connery col. 3, line 46-53.

Second set of rejection

14. Claims 1-6, 8,9, and 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Deb (U.S. 6,172,990) in view of Sarkissian (U.S. 6,789,116).

Regarding Claim 1, Deb discloses a method of evaluating an incoming data packet at a network switch port (see FIG. 2A, Media Access Controller MAC port 150), the method comprising:

storing a plurality of templates (see FIG. 3A, user defined instruction sets 300 for examining packet format) configured for identifying respective data formats (see FIG. 4A, step 402, or FIG. 4B, step 420; program/load/stores desired type of data structure format for examining packet formats; see col. 16, lines 16-20; see col. 17, lines 60-67; col. 11, line 13-

14 and 43-48), each template having at lease one term configured for comparing a corresponding prescribed value (see col. 17, lines 50-67; flags, fields, pointer, hashed data) to a corresponding selected byte of the incoming data packet (see Fig. 3B, Analyzing Computer 337; and col. 13, line 4-35, noted that the user defined contents are loaded/stored into the CAM 334, RAM 302, Comparators 336. A selected word/field (i.e. a word/field = a byte) from the incoming packet stream is being compared to the user defined instruction set);

comparing to the selected byte (see Fig. 3B, Analyzing Computer 337; col. 13, line 36-50; col. 4, lines 3-16, col. 13, lines 35-50), the terms that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port (see col. 13, line 51 to col. 14, line 6; noted that a selected word from an in-coming packet is being compared/examined according to the user defined data structure type and contents; also see Fig. 4A method step 404 and 406; Fig. 4C; and col. 25, line 31-32);

generating a comparison result that identifies the incoming data packet, based on the comparisons of the terms to the data bytes of the entire packet received by the network switch port (see col. 14, line 13-61; noted that compared/computed output (i.e. in the form of the data structure) between selected word and the user defined data instructions are produced at the analyzing computer. The data structure contains the information regarding computed results such as TCP header field, IP header field, the source and destination addresses, and etc. Therefore, it is clear the "comparison result" is the "computed output"); also see Fig. 4A method step 408; and col. 25, line 33-35; and

generating a frame tag (see Fig. 8, Encapsulation header 804 is a frame tag); based on the comparison result as soon as a last bit of the data packet is received at the network

switch port (see col. 29, line 10-29; and col. 21, line 30-46; noted that a new tag header is generated with an appended index (i.e. the user defined instructions) based upon compared/determined output data; see col. 4, lines 49-63. Thus, it is clear that a header is must be generated according to the examined/compared results on-the-fly according to the line rate. see Fig. 4A method step 408; and see col. 26, line 41-44).

Deb does not explicitly disclose min terms and simultaneously comparing. However, Sarkissian teaches storing a plurality of templates (see col. 3, lines 42-47; a reference register; see FIG. 3, Pattern parse and extraction database 308 and complier and optimizer 310, and Database of flows 324 stores references/masks/templates/flows; see col. 9, lines 6-7; see col. 10, lines 39-46) configured for identifying respective data formats (see FIG. 3, Analyzer and recognize pattern information PAR 304; see col. 8, lines 65 to col. 9, lines 6), each template having at least one min term (see col. 3, lines 42-45; a reference string of NR units; see FIG. 2, Key/signature 210; see col. 10, lines 1-14; see col. 29, lines 25-45) configured for comparing a corresponding prescribed value (see FIG. 2, S1,C1, or p1 of Key signature 210/212) to a corresponding selected byte of the incoming data packet (see FIG. 3, packet 302; see FIG. 2, S1,C1, or p1 byte of packet 206; note an S1 byte of the received packet is compared with S1 byte of the key; see col. 29, lines 25-45; see col. 30, lines 1-40; see col. 3, lines 46-55); also see FIG. 5, step 508, see col. 15, lines 65 to col. 16, lines 6; see FIG. 8, step 807.

simultaneously comparing, (see FIG. 3, a combined system of Complier and optimizer 310, PAR 304, EII 306, state processor instruction database 326 and lookup from records 314; see col. 3, lines 55-65; parallel comparing) to the selected byte (see col. 29, lines

25-45; see col. 30, lines 1-40; the selected byte S1, C1, or p1 is simultaneously match/compared; see col. 10, lines 38-45), the min terms (see FIG. 2, S1, C1, and p1 of key 210/212) that correspond to the selected byte immediately upon receipt of the selected byte (see FIG. 3, PAR 304, EII 306, and Lookup from records 314 process the comparing S1,C1, and p1 to the key upon immediately; see col. 9, lines 50-65, col. 10, lines 5-15, 38-45);

generating a comparison result (see FIG. 3, New flow record 316 (yes) or more classification on old flow (no)) that identifies the incoming data packet), based on the comparisons of the min terms (see FIG. 2, S1, C1, p1 of key 210/212) to the data bytes of the entire packet received (see FIG. 2, S1, C1, p1 of packet 206); see col. 11, lines 23-42. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide key/signature terms and simultaneously comparing process in the packet monitoring, as taught by Sarkissian in the system of Deb, so that it would meet any network speed requirement by performing real-time elucidation of packets communication including classification according to protocol and application program; see Sarkissian col. 1, line 40-42, see col. 3, lines 23-40.

Regarding Claim 11, Deb discloses a network switch port filter configured for evaluating an incoming data packet, comprising:

a term memory (see Fig. 3; RAM 302, CAM 334 and Comparators 336 which stored Programming instruction set) configured for storing term values (see FIG. 4A, step 402, or FIG. 4B, step 420; program/load/stores desired type of data structure format for examining packet formats; see col. 16, lines 16-20; see col. 17, lines 60-67; col. 11, line 13-14 and 43-48), each term value (see col. 17, lines 50-67; flags, fields, pointer, hashed data) stored based

on a location of a corresponding selected byte of the incoming data packet for comparison (see Fig. 3B, Analyzing Computer 337; and col. 13, line 4-35, noted that the user defined contents are loaded/stored into the CAM 334, RAM 302, Comparators 336. A selected word/field (i.e. a word/field = a byte) from the incoming packet stream is being compared to the user defined instruction set), a min term portion specifying a corresponding comparison operation (see Fig. 3A, Data structure content field and col. 11, line 43-56; noted that a data structure content field (i.e. pointer, data, and/or other) is used to identify what determination/operation will be performed), and an equation identifier field that specifies templates that use the corresponding min term (see Fig. 3A, Data structure type field and col. 11, line 43-56; noted that a data structure type field (i.e. a Standard data, flag, or other fields) is used to identify what type of the user defined instruction will be used for comparison operation);

comparing to the selected byte (see Fig. 3B, Analyzing Computer 337; col. 13, line 36-50; col. 4, lines 3-16, col. 13, lines 35-50), the terms that correspond to the selected byte immediately upon receipt of the selected byte by the network switch port (see col. 13, line 51 to col. 14, line 6; noted that a selected word from an in-coming packet is being compared/examined according to the user defined data structure type and contents; also see Fig. 4A method step 404 and 406; Fig. 4C; and col. 25, line 31-32);

a min term generator (see Fig. 3B, Analyzing Computer 337; noted that analyzing computer performs both equation core and min term generating function) configured for comparing a byte of the incoming data packet immediately upon receipt of the incoming data byte, with the terms that correspond to the received byte (see col. 13, line 51 to col. 14, line

6; noted that a selected word from an in-coming packet is being compared/examined according to the user defined data structure type and contents; also see Fig. 4A method step 404 and 406; Fig. 4C; and col. 25, line 31-32) and generating respective term comparison result ((see col. 14, line 13-61; noted that compared/computed output (i.e. in the form of the data structure) between selected word and the user defined data instructions are produced at the analyzing computer; Also see Fig. 4A method step 408; and col. 25, line 33-35.)

an equation core (see Fig. 3B, Analyzing Computer 337; noted that analyzing computer performs both equation core and min term generating function) configured for generating a frame tag (see Fig. 8, Encapsulation header 804 is a frame tag) identifying the incoming data packet based on the min term comparison results relative to the templates (see col. 29, line 10-29; and col. 21, line 30-46; noted that a new tag header is generated with an appendix index (i.e. the user defined contents) based upon compared/computed output data).

Deb does not explicitly disclose min terms and simultaneously comparing. However, Sarkissian teaches a min term memory for storing min term values (see col. 3, lines 42-47; a reference register; see FIG. 3, Pattern parse and extraction database 308 and complier and optimizer 310, and Database of flows 324 stores references/masks/templates/flows; see col. 9, lines 6-7; see col. 10, lines 39-46), each min term value (see col. 3, lines 42-45; a reference string of NR units; see FIG. 2, Key/signature 210; see col. 10, lines 1-14; see col. 29, lines 25-45) stored based on a location of corresponding selected byte of the incoming data packet for comparison, (see FIG. 3, packet 302; see FIG. 2, S1,C1, or p1 byte of packet 206; note an S1 byte of the received packet is compared with S1 byte of the key; see col. 29, lines 25-45; see col. 30, lines 1-40; see col. 3, lines 46-55); also see FIG. 5, step 508, see col. 15, lines 65

to col. 16, lines 6; see FIG. 8, step 807), a min term portion specifying a corresponding comparison operation (see FIG. 2, S1,C1,p1 portion of Key specifies a corresponding operation), and an equation identifier field (see FIG. 2, Key field (i.e. Key –1) that specifies templates that use the corresponding min term (see col. 30, lines 35-41; Key number field denotes different application templates);

a min term generator (see FIG. 3, a combined system of Complier and optimizer 310, PAR 304, EII 306, state processor instruction database 326 and lookup from records 314) configured for simultaneously comparing (see col. 3, lines 55-65; parallel comparing) a byte of the incoming data packet (see col. 29, lines 25-45; see col. 30, lines 1-40; the selected byte S1,C1, or p1 is simultaneously match/compared; see col. 10, lines 38-45), immediately upon receipt of the incoming data byte, with the min terms (see FIG. 2, S1, C1, and p1 of key 210/212) that correspond to the received byte (see FIG. 3, PAR 304, EII 306, and Lookup from records 314 process the comparing S1,C1, and p1 to the key upon immediately; see col. 9, lines 50-65, col. 10, lines 5-15, 38-45) and generating respective min term comparison results (see FIG. 3, New flow record 316 (yes) or more classification on old flow (no)) that identifies the incoming data packet) see col. 11, lines 23-42. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide key/signature terms and simultaneously comparing process in the packet monitoring, as taught by Sarkissian in the system of Deb, so that it would meet any network speed requirement by performing real-time elucidation of packets communication including classification according to protocol and application program; see Sarkissian col. 1, line 40-42, see col. 3, lines 23-40.

Regarding Claim 2, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 above. Deb discloses terms. Sarkissian discloses comparing in parallel (see col. 3, lines 55-61) comparing and min terms. Deb further discloses loading the terms (see Fig. 4A, step 402; and col. 16, line 15-19) corresponding to a first of the data bytes into a term generator (see Fig. 3B, Analyzing Computer 337; see Fig. 4A, step 404; and see col. 16, line 24-39; noted that the user defined instruction/context are loaded/stored in the RAM, CAM, and Comparators associating to the initial byte of an incoming packet); comparing in the terms loaded in the term generator with the first of the data bytes (see Fig. 4A, step 406; col. 16, line 47-52; since the incoming packet are stored in the register in order, the first word (i.e. word count "0") is used when examining/comparing the received packet); and outputting comparison results for the min terms loaded in the min term generator to an evaluation core (see FIG. 3B, a combined system of Analyzing Computer 337, Mux 318, and Data Structure Register 316; see Fig. 4A, step 408; col. 16, line 55-61; noted that the output from compared/determined data are transferred to a combined system of Mux 318 and data transfers register 316).

Sarkissian also discloses loading the min terms corresponding to a first of the data bytes (see col. 3, lines 42-47; a reference register; see FIG. 3, Pattern parse and extraction database 308 and complier and optimizer 310, and Database of flows 324 stores references/masks/templates/flows; see col. 9, lines 6-7; see col. 10, lines 39-46) into a min term generator see FIG. 3, a combined system of Complier and optimizer 310, PAR 304, EII 306, state processor instruction database 326 and lookup from records 314; see col. 9, lines 6-7; see col. 10, lines 39-46), comparing in the min terms loaded in the min term generator with

the first of the data bytes (see FIG. 3, packet 302; see FIG. 2, S1,C1, or p1 byte of packet 206; note an S1 byte of the received packet is compared with S1 byte of the key; see col. 29, lines 25-45; see col. 30, lines 1-40; see col. 3', lines 46-55); and outputting comparison results for the min terms loaded in the min term generator (see FIG. 3, New flow record 316 (yes) or more classification on old flow (no)) that identifies the incoming data packet) see col. 11, lines 23-42) to an evaluation core (see FIG. 3, More classification 320, protocol and state identification 318; see col. 11, lines 23-46).

Regarding Claim 3, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 above. Deb discloses loading the terms corresponding to a second of the data bytes, contiguously following the first of the data bytes, into the term generator (see Fig. 4C; col. 25, line 39-58; and col. 13, line 34-50; noted that a second of the data byte is stored contiguously according to the method defined in Fig. 4C).

Regarding Claim 4, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 above. Deb discloses outputting the frame tag to a switch fabric (see Fig. 2A, a combined system of Tx micro-RISC Stream Processor 114a and Switch Table Lookup 806) configured for selectively switching the incoming data packet based on the corresponding frame tag (see col. 21, line 14-61; and col. 22, line 46-57; noted that the user defined appendix is append to an incoming packet, transferred to a lookup switch table).

Regarding Claim 5, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 above. Deb discloses the storing step includes storing each term in a memory as a table entry (see Fig. 3A, RAM 302, CAM 334, and Comparators 336), each table entry having a location in the memory based on a location of the corresponding

selected byte in the incoming data packet (see col. 13, line 24-35; noted that the user instructions are resident in word count 308 is configured to identify a desired word count in an in-coming packet), the table entry including a term expression portion specifying the corresponding prescribed value and a comparison operand (see Fig. 3A, Data structure content field; and col. 11, line 43-56; noted that a data structure content field (i.e. pointer, data, and/or other) is used to identify what determination/operation will be performed), and a equation identifier field that specifies the templates that use the corresponding term (see Fig. 3A, Data structure type field and col. 11, line 43-56; noted that a data structure type field (i.e. a Standard data, flag, or other fields) is used to identify what type of the user defined instruction will be used for comparison operation.)

Sarkissian also discloses storing each min term in a memory as a table entry (see FIG. 2, S1, C1, p1 and port number entries in the Key/signature memory), each table entry having a location in the memory (see FIG. 2, location of source address S1, destination address C1, protocol p1, or port number field 243 of the KEY 1) based on a location of the corresponding selected byte in the incoming data packet (see FIG. 2, corresponding byte location of source address S1, destination address C1, protocol p1, or port number field 243 of the incoming packet 206); see col. 29, lines 25-60); the table entry including a min term expression portion specifying the corresponding prescribed value (see FIG. 2, S1,C1, p1 value of packet) and a comparison operator filed ((see FIG. 2, S1,C1,p1 filed in the Key specifies a corresponding operation for each value), and an evaluation portion having an equation identifier field (see FIG. 2, Key field (i.e. Key –1)) that specifies the templates that use the corresponding min term (see col. 30, lines 35-41; Key number field denotes different application templates).

Regarding Claim 6, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 and 5 above. Deb discloses temporarily storing results of the comparisons of the terms to the selected bytes of the incoming data packet (see col.14, line 28-30; noted that the compared/computed output data are stored in a data structure having a pointer to the currently selected word) detecting at least one matched template from the plurality of templates based on the results of the comparisons of the terms and generating the comparison result based on the detected at least one matched template (see col. 13, line 17-23; a match signal is produced (i.e. match found) after matching according to the CAM 334 look up table which stored plurality of user defined instructions and the corresponding entry (i.e. compared/determined output data) are outputted.)

Regarding Claim 8, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 above. Deb discloses the first of the data bytes corresponds to a first of the data bytes of a packet having a prescribed format, the comparing step including evaluating the selected data byte relative to a beginning of the packet having the prescribed format (see col. 13, line 51 to col. 14, line 6; noted that the determination is performed by the analyzing computer from the initial word (i.e. beginning of the packet) and continues consecutively). Sarkissian discloses the first of the data bytes (see FIG. 2, S1 of packet 206) corresponds to a first of the data bytes of a packet having a prescribed format (see FIG. 2, S1 of key 210), the simultaneously comparing step including evaluating the selected data byte (see FIG. 2, S1, C1, or p1 of packet 206) relative to a beginning of the packet having the prescribed format (see FIG. 2, S1, C1, or p1 of Key1 210); see col. 3, lines 55-65; see col. 9, lines 50-65, col. 10, lines 5-15, 38-45.

Regarding Claim 9, Deb discloses the prescribed format is Internet protocol (IP) format (see Fig. 9, IP switching 3 – IP header 910 and data 906; and col. 20, line 54-60).

Regarding Claim 12, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 11 above. Deb discloses a frame identifier (see Fig. 3B, CAM 334) configured for identifying a type of layer 2 packet (see col. 9, line 18-23; and col. 13, line 16-23; noted that MAC layer is a Layer 2, and the CAM identifies by utilizing a look up table to process for each word type before passing over to the higher layers), the selected byte of the incoming data packet determined based on the identified type of layer 2 packet (see col. 13, line 59-63; col. 19, line 59 to col. 20, line 3; and Fig. 5A and 5B; noted that selected word of the incoming packet is determined according to the user defined instructions for different type of layer 2.)

Regarding Claim 13, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 11 above. Deb discloses the location of each stored term value is relative to a beginning of an IP frame (see Fig.9, IP header 910; and col. 20, line 54-60; noted that received packet header is an IP header, and therefore, it is an IP frame.) within the layer 2 packet (see col. 13, line 33-50; the user defined instructions are resident in word count 308, and it is configured to identify a desired word count in an in-coming packet. Each time a new packet is received by micro-RISC stream processor 114a, a word counter 307 will reset to "0", and then word counter 307 begins sequentially counting each word that is received into pipeline register stages 323 from data path 115a. Therefore, each stored the user defined instruction is relative to a beginning of an IP frame).

Regarding Claim 14, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 11 above. Deb discloses a min term controller (see Fig. 3B, Execution Logic 312) configured for fetching the min terms from the min term memory corresponding to a selected byte of the IP frame within the incoming data packet (see col. 15, line 45-49 and col.14, line 12-19; noted that the execution logic unit is preferably designed to control the examination of the received packet by the analyzing computer.)

Regarding Claim 15, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 11 above. Deb discloses the equation core generates the frame tag at a wire rate of the incoming data packet and prior to an end of the incoming data packet (see col. 11, line 21-42; col. 21, line 30-46; and col. 29, line 10-27; Noted that both Tx and Rx micro-RISC stream processors are considered as one system. Therefore, the processor has a capability of generation a frame tag, which encapsulates to the data frame.)

15. Claim 7 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Deb in view of Sarkissian, as described above in claims 1 and 11, and further in view of Connery (U.S. Patent 6,570,884).

Regarding claim 7, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 1 above. Deb teaches generating a final tag value after matching the words to the user-defined instructions. Deb further teaches prioritizing the incoming data traffic at the receiver utilizing various buffers (see col. 10, line 59-65) and transmitting different type of traffic according to the priority (see col. 7, line 33-38). Deb also teaches the user-defined instruction, and the user is able to define/prioritize the traffic type

(i.e. voice vs. data or IP vs. SNMP) for by utilizing instruction. Neither Deb nor Sarkissian explicitly disclose resolving a priority of templates to one final template when more than one template matches the incoming data packet. However, Connery discloses resolving a priority of templates (see Fig. 3, Pattern Match units 1-4) to one final template when more than one template matches the incoming data packet (see col. 7, line 52-62; noted that when there are multiple matching of patterns (i.e. more than one matching to the defined pattern), the processor determines the final matching format). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide hardware pattern matching logic that resolve a priority of templates to one final template, as taught by Connery, in the combined system of Deb and Sarkissian, so that it would provide hardware pattern matching logic which supports pattern matching at the speed of the incoming packet stream, and signals the embedded processor when a packet having one of the plurality of variant formats is detected, thereby, minimizes the probability of faulty matches; see Connery col. 3, line 46-53.

Regarding claim 16, the combined system of Deb and Sarkissian discloses all of the limitations as recited in claim 11 above. Deb teaches generating a final tag value after matching the words to the user-defined instructions. Deb further teaches prioritizing the incoming data traffic at the receiver utilizing various buffers (see col. 10, line 59-65) and transmitting different type of traffic according to the priority (see col. 7, line 33-38). Deb also teaches the user-defined instruction, and the user is able to define/prioritize the traffic type (i.e. voice vs. data or IP vs. SNMP) for by utilizing instruction. Neither Deb nor Sarkissian explicitly disclose a priority device configured for resolving a priority of templates to one

final frame template when more than one template matches the incoming data packet. However, Deb teaches a priority device (see Fig. 3, Processor 220) configured for resolving a priority of templates (see Connery '884 Fig. 3, Pattern Match units 1-4) to one final frame template when more than one template matches the incoming data packet (see col. 7, line 52-62; noted that when there are multiple matching of patterns (i.e. more than one matching to the defined pattern), the processor determines the final matching format). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide hardware pattern matching logic that resolve a priority of templates to one final template, as taught by Connery, in the combined system of Deb and Sarkissian, so that it would provide hardware pattern matching logic which supports pattern matching at the speed of the incoming packet stream, and signals the embedded processor when a packet having one of the plurality of variant formats is detected, thereby, minimizes the probability of faulty matches; see Connery col. 3, line 46-53.

16. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Deb in view of Sarkissian, as described above in claims 1 and 11, and further in view of Bellenger (U.S. 5,949,786).

Regarding Claim 10, the combined system of Deb and Sarkissian discloses the step of generating the comparison result based on the detected at least one matched template includes: identifying for each of the min terms compared with the incoming data packet and specifying a unique result for a selected group of the templates; and generating the comparison result by having the detected at least one matched template as described above in

Claim 1, 5, and 6. Neither Deb nor Sarkissian explicitly discloses an equation, each equation specifying a unique result for a selected group of the templates; and generating the comparison result by the equation. However, Bellenger discloses an equation (see col. 9, lines 64-65; see col. 10, lines 16-20; a template register equation (i.e. FF-F0-00-00-00-00-00-00 for bridging)), each equation specifying a unique result for a selected group of the templates; and generating the comparison result by the equation (see col. 7, lines 10 to col. 8, lines 20; see also Fig. 6). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a template equation to generate the comparison/matching result, as taught by Bellenger, in the combined system of Deb and Sarkissian, so that it would provide a fast network flow switching technique which adaptable to wide variety of protocols and improves performance of multiprotocol network switches and routers; see Bellenger col. 2, line 35-40.

Response to Arguments

17. Applicant's arguments with respect to claims 1-16 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ian N Moore whose telephone number is 571-272-3085. The examiner can normally be reached on M-F: 8:30 AM - 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ken Vanderpuye can be reached on 571-272-3078. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

INM
12/8/04

BRIAN NGUYEN
PRIMARY EXAMINER